AMENDMENTS TO THE CLAIMS:

Please amend the claims as follows:

1 – 46. (Cancelled)

47. (Currently Amended) A compute node capable of operating as part of a distributed system, comprising:

memory; and

a processor configured to:

access the memory to perform a process in a distributed computation running on the distributed system;,

~~the processor being further configured to~~ record a first set of memory locations modified by the processor during a first checkpoint interval ~~and~~;

create a checkpoint from the contents of the first set of memory locations by storing the contents of the first set of memory locations in a certain order [[,]] while recording a second set of memory locations modified by the processor during a second checkpoint interval;

write protect the first set of memory locations before modifying the second set of memory locations;

store the contents of a memory location from the first set earlier than it would otherwise be stored when the processor needs to modify the memory location during the second checkpoint interval; and

remove the write protection for a memory location from the first set when the processor needs to modify the memory location during the second checkpoint interval after the contents of the memory location has been stored.

48. (Cancelled)

49. (Currently Amended) The compute node of claim 4748 wherein the processor is further configured to suspend the process between the first and second checkpoint intervals, the processor being further configured to write protect the first set of memory locations while the process is suspended.

50. (Currently Amended) The compute node of claim 4947 wherein the processor is further configured to ~~enter~~ execute a barrier operation following the completion of write protecting the first set of memory locations and ~~end~~exit the barrier operation before resuming the process during the second checkpoint interval.

51. (Cancelled)

52. Cancelled)

53. (Currently Amended) The compute node of claim 4752 wherein the processor is further configured to remove the write protection for the memory location after the contents of the memory location has been stored.

54. (Previously Presented) The Compute node of claim 47 further comprising a checkpoint file, wherein the processor is further configured to create the checkpoint by storing the contents of the first set of memory locations to the checkpoint file.

55. (Currently Amended) The compute node of claim 54 wherein the processor is further configured to remove the record of a memory location from the first set after the contents from the memory location [[is]]area stored in the checkpoint file.

56. Previously Presented) The compute node of claim 47 wherein the processor is further configured to configured to create the checkpoint by storing the contents of the first set of memory locations to non-volatile storage.

57. Previously Presented) The compute node of claim 47 wherein the processor is further configured to store in the memory a copy of each message output from the compute node during the process until an acknowledgement is received, and output each message copied in the memory that does not receive an acknowledgement, and

wherein the processor is further configured to receive messages during the process, and output an acknowledgement for each message received, the processor being further configured to recognize and discard duplicate messages received by the compute node, and for each duplicate message, output an acknowledgement.

58-65. (Cancelled)

66. (Currently Amended) Computer readable media embodying a program of instructions executable by a processor to perform a method of creating a checkpoint for a process in a distributed computation running on a distributed system, the method comprising:

recording a first set of memory locations modified by the process during a first checkpoint interval; ~~and~~

creating a checkpoint from the contents of the first set of memory locations, while recording a second set of memory locations modified by the process during a second checkpoint interval; and

write protecting the first set of memory locations before the process modifies the second set of memory locations;

wherein the checkpoint is created by storing the contents of the first set of memory locations in a certain order, the method further comprising:

storing the contents of a memory location from the first set earlier than it would otherwise be stored when the process needs to modify the memory location during the second checkpoint interval; and

removing the write protection for a memory location from the first set when the process needs to modify the memory location during the second checkpoint interval after the contents of the memory location has been stored.

67. (Cancelled)

68. (Currently Amended) The computer readable media of claim 66~~7~~ wherein the method further comprises suspending the process between the first and second checkpoint intervals, and wherein the first set of memory locations are write protected while the process is suspended.

69. (Currently Amended) The computer readable media of claim 66~~7~~ wherein the method further comprises entering a barrier following the completion of write protecting the first set of memory locations and exiting the barrier before resuming the process during the second checkpoint interval.

70. Cancelled)

71. (Cancelled)

72. (Currently Amended) The computer readable media of claim 66~~71~~ wherein the method further comprises removing the record of the memory location and the write protection for the memory location, after the contents of the memory location has been stored.

73. (Previously Presented) The computer readable media of claim 66 wherein the checkpoint is created by storing the contents of the first set of memory locations to a checkpoint file.

74. (Previously Presented) The computer readable media of claim 73 wherein the method further comprises removing the record of a memory location from the first set after the contents from the memory location is stored in the checkpoint file.

75. Previously Presented) The computer readable media of claim 66 wherein the checkpoint is created by storing the contents of the first set of memory locations to non-volatile storage.

76. (Currently Amended) The computer readable media of claim 66 wherein the process is performed by a compute node in the distributed system, the method further comprising storing in the memory a copy of each message output from the compute node during the process until an acknowledgement is received, and outputting each message copied in the memory that does not receive an acknowledgement, and wherein the method further comprises receiving messages during the process, outputting an acknowledgement for each message received, recognizing and discarding duplicate messages received by the compute node[[,]] and, for each duplicate message, outputting an acknowledgement.

77-87 (Cancelled)

88. (Currently Amended) A method of creating a checkpoint for a process in a distributed computation running on a distributed system, the method comprising:

recording a first set of memory locations modified by the process during a first checkpoint interval; ~~and~~

creating a checkpoint from the contents of the first set of memory locations by storing the contents of the first set of memory locations in a certain order, while recording a second set of memory locations modified by the process during a second checkpoint interval;

write protecting the first set of memory locations before the process modifies the second set of memory locations;

storing the contents of a memory location from the first set earlier than it would otherwise be stored when the process needs to modify the memory location during the second checkpoint interval; and

removing the write protection for a memory location from the first set when the process needs to modify the memory location during the second checkpoint interval after the contents of the memory location has been stored.

89. (Cancelled)

90. (Currently Amended) The method of claim 88~~9~~ further comprising suspending the process between the first and second checkpoint intervals, and wherein the first set of memory locations are write protected while the process is suspended.

91. (Currently Amended) The method of claim 88~~90~~ further comprising entering a barrier following the completion of write protecting the first set of memory locations and exiting the barrier before resuming the process during the second checkpoint interval.

92. (Cancelled)

93. (Cancelled)

94. (Currently Amended) The Method of claim 88~~93~~ further comprising removing the record of the memory location and the write protection for the memory location, after the contents of the memory location has been stored.

95. (Previously Presented) The method of claim 88 wherein the checkpoint is created by storing the contents of the first set of memory locations to a checkpoint file.

96. (Previously Presented) The method of claim 95 further comprising removing the record of a memory location from the first set after the contents from the memory location is stored in the checkpoint file.

97. (Previously Presented) The method of claim 88 wherein the checkpoint is created by storing the contents of the first set of memory locations to non-volatile storage.

98. (Previously Presented) The method of claim 88 wherein the process is performed by a compute node in the distributed system, the method further comprising storing in the memory a copy of each message output from the

compute node during the process until an acknowledgement is received, and outputting each message copied in the memory that does not receive an acknowledgement, and wherein the method further comprises receiving messages during the process, outputting an acknowledgement for each message received, recognizing and discarding duplicate messages received by the compute node, and for each duplicate message, outputting an acknowledgement.

99-109 (Cancelled)